ICS-5 实验报告

李浩然　2025 年 12 月 16 日

1　实验目标与背景

在 LC-3 体系下，用户程序通过 TRAP 调用系统服务（如 IN、PUTS、PUTSP 等）访问 I/O。若内核实现的 ISR 对来自用户的输入未做长度检查，则可能发生类似 C 语言中 strcpy 导致的缓冲区溢出，从而覆盖内核代码或其它敏感数据，进而操纵控制流并实现提权。本实验目标为：在给定的脆弱 ISR 与已存在于 x4000 的“恶意程序”的条件下，设计并提交一个用户态程序，使得最终能够在管理/特权态执行 x4000 的恶意程序。

2　漏洞描述

题中 ISR 的行为如下：

- 将 R0/R1/R2 保存；
- 将 R1 设为 PROMPT_ADDR（0x032C）；
- 以 word 为单位读取 R0 指向的内存并写入 R1 指向的位置（LDR/STR），每次复制后 R0、R1 +1；
- 当读取到的 word 为 0（0x0000）时终止；
- 恢复寄存器并 RTI 返回。

因此：

- 首先会把用户提供的前 20 words 覆写 PROMPT_ADDR（0x032C..0x033F）；
- 随后继续写入，会按顺序覆盖紧随其后的 PUTSP 的实现代码，直到遇到 0；
- 缺乏边界检查导致用户数据可以覆盖内核代码（code overwrite）。

3　攻击思路

核心思路为在 PUTSP 的入口处写入一个短的跳转序列，该序列会在内核/特权模式下把控制流转到 x4000（已存在恶意程序）。

LD R0, #1 ; 将紧跟 LD 指令之后的常量（即 x4000）装入 R0

JMP R0 ; 间接跳转到 R0 中的地址

.FILL x4000 ; 常量（目标地址）

.FILL x0000 ; 终止符，保证 ISR 停止复制

payload 格式为：

- 前 20 words：任意 filler（占位，覆盖 PROMPT 区）
- 紧接若干 words：上面的跳转序列（每项用.FILL 写入 16-bit word）
- 末尾一个 x0000 作复制终止符

当 TRAP x30 调用脆弱 ISR 时，ISR 把 payload 复制到 PROMPT 与随后内存，覆盖到 PUTSP 起始处；随后用户触发 TRAP x24（PUTSP）时，系统会在特权态执行 PUTSP，而 PUTSP 起始的指令将实现跳转到 x4000，从而在特权态运行 x4000 的恶意程序，实现提权。

4 结论

本实验成功演示了在 LC-3 环境下通过对脆弱 ISR 进行“缓冲区式”溢出写入实现的代码覆盖与提权流程：我们将 PUTSP 的起始指令替换为跳转序列并最终在特权态执行了 x4000 处的恶意程序。该实验强调了内核/系统服务对用户输入进行边界与权限检查的重要性，并提供了调试与防御思路。

A 攻击者程序

```
        .ORIG x3000
        LEA R0, PAYLOAD
        TRAP x30
        TRAP x24
        HALT

PAYLOAD
        ; 20x filler
        .FILL x0041
        .FILL x0041
        .FILL x0041
        .FILL x0041
        .FILL x0041
        .FILL x0041
        .FILL x0041
        .FILL x0041
        .FILL x0041
        .FILL x0041
        .FILL x0041
        .FILL x0041
        .FILL x0041
        .FILL x0041
        .FILL x0041
        .FILL x0041
        .FILL x0041
        .FILL x0041
        .FILL x0041
        .FILL x0041

        .FILL x2001
        .FILL xC000
        .FILL x4000
        .FILL x0000

        .END
```